IN THE SPECIFICATION

Please amend the Specification as follows:

Please **replace** the paragraph beginning at **Page 5, line 12**, with the following paragraph:

1) An overlay network of alternate routing mechanisms is constructed on top of the existing Internet routing mechanisms to find and exploit available resources. The overlay routing mechanism is completely transparent and separate from the Internet routing protocols and is preferably deployed throughout some small, but widely distributed, portion of the Internet as a distributed user application. Figure 1 exemplifies the concept. Nodes 100 and 160 are, respectively, source and destination nodes for an intended communication on a network such as the Internet. These nodes are connected to the underlying network via transmission links 110 and 160 ~~170~~, respectively. Nodes 140a-n (connected to the underlying network via links 145a-n) represent other network nodes, and might potentially be nodes that are utilized in a default communication path between node 100 and node 170, depending on the routing mechanisms of the network. Overlay network nodes 130a-n utilize existing network transmission lines and infrastructure, via network links 135a-n, to create a virtual topology. The overlay network preferably includes a number of computing devices such as nodes 130a-n that cooperate to provide forwarding paths overlaid over an underlying network. Overlay network nodes preferably communicate using existing, established Internet protocols and thus do not require any modifications to current standards. Each overlay node 130 preferably includes overlay path module 150, and either the source or destination node similarly includes overlay path module 120; these components are programmed and operable to combine available IP protocols in order to provide additional functionality for exploiting overlay routing when it is advantageous to do so, as described below in detail.

Please **replace** the paragraph beginning at **Page 7, line 16**, with the following paragraph:

Our invention preferably provides on-demand routing, discovering and adding useful forwarding paths through the overlay network only when needed. This avoids having to pre-compute and record all possible forwarding paths in advance, and advantageously uses the default Internet routing mechanism for bootstrapping and default operations. More particularly, the preferred embodiment of our invention creates a new forwarding path from endpoint A to endpoint B only when: (1) an end-to-end communication is requested between A and B (per step 200 of Fig. 2), and (2) a path is ~~id~~ discovered through the overlay routing network that provides better performance than the default Internet route (per steps 210-215 of Fig. 2).

Please **replace** the paragraph beginning at **Page 7, line 26**, with the following paragraph:

Therefore, the discovery of an overlay forwarding path preferably starts with monitoring one or more cost/performance metrics of interest for the data communications that are carried out on the default Internet routing path. Such monitoring would most typically be performed at a gateway router or the source endpoint, node 100. Module 120 ~~110~~ employs a predetermined cost function that combines the monitored metrics and detects end-to-end communications that do not meet specific predetermined requirements. For such communications, the detection process would extract from the monitoring operations (1) the source address A, (2) the destination address B and (3) the cost of the data communication from A to B. Computation of cost information is discussed further below. This information is then used in the process of on-demand forwarding path discovery, as discussed below.

Please **replace the paragraph** beginning at **Page 8, line** 7, with the following paragraph:

Source node 100 (as well as any of the routers on the default Internet forwarding path) can potentially discover end-to-end communications that do not meet specific requirements. In that event, in order to initiate steps 220-225, module <u>120</u> ~~110~~ sends a query to the overlay network nodes 130 to determine if the overlay network is capable of offering a better forwarding path. The query is preferably sent to a specified number ("q") of the overlay network routers 130, depending on the configuration. In a relatively simple embodiment, each of the q forwarding path query messages preferably includes: (1) a destination address B, (2) a source address A, and (3) an identifier for a predefined cost function F. In the example illustrated in Fig. 1, source A is node 100, and destination B is node 160. Cost function F is preferably drawn from a set of network communication performance metrics such as delay, throughput, jitter or loss, in accordance with the practitioner's priorities and needs.

Please **replace the paragraph** beginning at **Page 8, line 20** (running to pg. 9, line 3) with the following paragraph:

When each of the q overlay network nodes 130i receives a forwarding path query, it performs step 220 and measures the assigned cost function F with respect to communications transmitted to destination address B from overlay node, yielding the value F(B,i). F(B,i) is measured for a default network path from the ith overlay node to destination B. In this simple embodiment, the querying node's module 120 ~~110~~ receives a single reply from each of the q overlay network routers queried. The querying node at any time during the reception of the replies may decide to pick a particular forwarding path and ignore any additional query replies. In order to pick an optimized forwarding path, the querying node's module 120 ~~110~~ preferably combines the F(B) value in each reply with the cost function F(i,A) which measures the cost of communication to overlay node 130i from the querying node, once again along a default network path. As those of skill in the art will appreciate, the combining of cost functions may entail adding values (as where the cost metric is delay) or calculating the minimum value (as for bandwidth), or in general may involve a complex parameterized combination of the cost functions. In any case, at steps 230-235 module 120 ~~110~~ preferably uses the computed total costs for the alternative overlay paths and for the default path to select an optimized path for communication between source node 100 (A) and destination node 160 (B).

Please **replace the paragraph** beginning at **Page 13, line 20** (running to pg. 14, line 2) with the following paragraph:

For illustration, we will begin with a simple example, in which the message is one-way (no reply), and the alternative overlay path is a one-hop path (i.e., it goes through a single overlay node). In this example, the client at node 100 (or a client connected through gateway node 100 to the network) wishes to send a message on a network such as the Internet to destination node 160. In accordance with a preferred embodiment of the present invention, steps 210-240 are first performed, to discover an optimized overlay path for communicating with 160. Suppose this process determines that, at the present moment, an optimized path for sending a message to 160 (better than the default network path, at any rate) is to send packets from 100 to overlay node 130a, and then to forward them from 130a to 160. In other words, the desired path strategy is to send packets from 100 to 130a using the default network path for 100→130a, and then forward those packets from 130a to 160 using the default network path for 130a→160. At step 250, this transmission is actually carried out, as detailed in Fig. 5. At step 500, overlay software 120 ~~110~~ at node 100 addresses the packets to 130a, instead of 160, but also "encapsulates" or encodes the address of 160 in a predetermined format incorporated in the message. The message is then sent to overlay node 130a, at step 510, preferably by means of default network routing mechanisms. When 130a receives the packets, overlay software 150a decodes or de-encapsulates the encapsulated data, and finds the encoded "160" address. At step 520, module 150a of node 130a checks the overlay path information stored earlier at step 240 to identify the next node on the overlay forwarding path. Because, in this example, there are no more overlay nodes on the forwarding path, software 150a proceeds to step 530, and restores the original message with its destination address reset to node 160. Again, because this example involves no reply message, software 150a proceeds to step 580 and simply forwards the packets on to their final destination at node 160. In this way, the original message gets from client (or gateway) 100 to destination node 160, along an optimized non-default path passing through overlay node 130a. This is accomplished without any need to modify the established communications protocols of the underlying network (e.g., IP), and without any modification (or even awareness) of destination node 160.

Please **replace the paragraph** beginning at **Page 14, line 20** (running to pg. 14, line 2) with the following paragraph:

We next present a further example, involving a multi-hop overlay path; once again, the example treats a one-way communication. In this example, we assume that the process of steps 210-240 discovers an optimized path for transmitting messages from 100 to 160, passing through overlay nodes 130a and 130b. In other words, this time the desired path strategy is to send packets from 100 to 130a using the default network path for 100 → 130a, then forward those packets from 130a to 130b using the default network path for 130a → 130b, and finally to forward those packets from 130b to 160 using the default network path for 130b → 160. Once again, at step 500, overlay software 120 ~~110~~ at node 100 addresses the packets to 130a, and encapsulates the address of 160. The message is then sent to overlay node 130a, at step 510. When 130a receives the packets, overlay software 150a finds the encoded "160" address, and at step 520, software 150a of node 130a checks the overlay path information stored earlier at step 240 and identifies overlay node 130b as the next node on the overlay forwarding path. Following the flow of Fig. 5, module 150a loops back to step 510 and forwards the message to overlay node 130b, where module 150b performs similar functionality. This time, at step 520, module 150b determines that there are no more overlay nodes on the forwarding path, and thereupon (at step 530) restores the original message with its destination address reset to node 160. Because this example again involves no reply message, software 150b proceeds to step 580 and forwards the packets on to their final destination at node 160. In this way, the original message gets from client (or gateway) 100 to destination node 160, along an optimized non-default path passing through overlay nodes 130a and 130b; and once again, this is accomplished without any need to modify the established communications protocols of the underlying network.

Please **replace the paragraph** beginning at Page 14, line 29 to page 15, line 23 with the following:

As a third example, we will now consider the case of a message that requests a return reply (such as an http request to get a file), once again in the context of the multi-hop forwarding path through overlay nodes 130a and 130b as in the previous example. In this scenario, our preferred embodiment operates in the same manner as in the previous example, until module 150b reaches step 535 and determines that the message does indeed request a return reply from the destination node 160. Following the flow in Fig. 5, at step 540 module 150b "masquerades" source information for the packets. In our preferred embodiment, the last overlay node on a forwarding path performs the task of masquerading, in order to allow bi-directional use of the overlay forwarding path. In the absence of masquerading, the reply sent by node 160 to node 100 would normally follow a return path using default network routing. In general, masquerading replaces the source address of IP packets with the address of the node executing the masquerade, and records enough information locally so as to be able restore the original source address if and when a replay IP packet is returned. In a preferred embodiment and in the context of a network like the Internet, module 150 of a masquerading node locally stores the original source address and the port from which it sent the packet (a port uniquely identifies which connections a node has with any other network node). At step 550, overlay node ~~135b~~ 130b sends the masqueraded message to destination node 160. If and when reply packets are sent from node 160, they will be addressed to overlay node ~~135b~~ 130b, because of the masqueraded source information. When the reply comes back on the appropriate port of node ~~135b~~ 130b, at step 570 module 150b retrieves the original source address for node 100 that was previously stored at step 540--which is the true intended destination of the reply message being handled--and constructs a reply message encapsulating the intended destination address of node 100. Returning to step 510, module 150b forwards the encapsulated message to the next overlay node on an optimized path to node 100, by accessing path information previously stored at step 240 (in this case, the path information is of course just the inverse of the optimized overlay path for communications being sent from source 100 to destination 160).